Question Answering on a Case Insensitive Corpus

Wei Li, Rohini Srihari, Cheng Niu, Xiaoge Li

Cymfony Inc.
600 Essjay Road
Williamsville, NY 14221, USA
{wei, rohini, cniu, xli}@cymfony.com

Abstract

Most question answering (QA) systems rely on both keyword index and Named Entity (NE) tagging. The corpus from which the QA systems attempt to retrieve answers is usually mixed case text. However, there are numerous corpora that consist of case insensitive documents, e.g. speech recognition results. This paper presents a successful approach to QA on a case insensitive corpus, whereby a preprocessing module is designed to restore the case-sensitive form. The document pool with the restored case then feeds the QA system, which remains unchanged. The case restoration preprocessing is implemented as a Hidden Markov Model trained on a large raw corpus of case sensitive documents. It is demonstrated that this approach leads to very limited degradation in QA benchmarking (2.8%), mainly due to the limited degradation in the underlying information extraction support.

1 Introduction

Natural language Question Answering (QA) is recognized as a capability with great potential. The NIST-sponsored Text Retrieval Conference (TREC) has been the driving force for developing this technology through its QA track since TREC-8 [Voorhees 1999] [Voorhees 2000]. There has been significant progress and interest in QA research in recent years [Pasca & Harabagiu. 2001] [Voorhees 2000].

In real-life QA applications, a system should be robust enough to handle diverse textual media degraded to different degrees. One of the challenges from degraded text is the treatment of case insensitive documents such as speech recognition results, broadcast transcripts, and the Foreign Broadcast Information Service (FBIS) sources. In the intelligence domain, the majority of archives consist of documents in all uppercase.

The orthographic case information for written text is an important information source. In particular, the basic information extraction (IE) support for QA, namely Named Entity (NE) tagging, relies heavily on the case information for recognizing proper names. Almost all NE systems (e.g. [Bikel *et al.* 1997], [Krupka & Hausman 1998]) utilize case-related features. When this information is not available, if the system is not re-trained or adapted, serious performance degradation will occur. In the case of the statistical NE tagger, without adaptation the system simply does not work. The degradation for proper name NE tagging is more than 70% based on our testing. The key issue here is how to minimize the performance degradation by adopting some strategy for the system adaptation.

For search engines, the case information is often ignored in keyword indexing and retrieval for the sake of efficiency and robustness/recall. However, QA requires fine-grained text processing beyond keyword indexing since, instead of a list of documents or URLs, a list of candidate answers at phrase level or sentence level is expected to be returned in response to a query. Typically QA is

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE
2003

2. REPORT TYPE

3. DATES COVERED
00-00-2003 to 00-00-2003

4. TITLE AND SUBTITLE
Question Answering on a Case Insensitive Corpus

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Cymfony Inc,600 Essjay Road,Williamsville,NY,14221

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE		**10**	
unclassified	**unclassified**	**unclassified**			

supported by Natural Language Processing (NLP) and IE [Chinchor & Marsh 1998] [Hovy *et al.* 2001] [Srihari & Li 2000]. Examples of using NLP and IE in Question Answering include shallow parsing [Kupiec 1993] [Srihari & Li 2000], deep parsing [Li *et al.* 2002] [Litkowski 1999] [Voorhees 1999], and IE [Abney *et al.* 2000] [Srihari & Li 2000]. Almost all state-of-the-art QA systems rely on NE in searching for candidate answers.

For a system based on language models, a feature exclusion approach is used to re-train the models, excluding features related to the case information [Kubala *et al.* 1998] [Miller *et al.* 2000] [Palmer *et al.* 2000]. In particular, the DARPA HUB-4 program evaluates NE systems on speech recognizer output in SNOR (Standard Normalized Orthographic Representation) that is case insensitive and has no punctuations [Chincor *et al.* 1998]. Research on case insensitive text has so far been restricted to NE and the feature exclusion approach [Chieu & Ng 2002] [Kubala *et al*. 1998] [Palmer *et al.* 2000] [Robinson *et al.* 1999]. When we examine a system beyond the shallow processing of NE, the traditional feature exclusion approach may not be feasible. A sophisticated QA system usually involves several components with multiple modules, involving NLP/IE processing at various levels. Each processing module may involve some sort of case information as constraints. It is too costly and sometimes impossible to maintain two versions of a multi-module QA system for the purpose of handling two types of documents, with or without case.

This paper presents a case restoration approach to this problem, as applied to QA. The focus is to study the feasibility of QA on a case insensitive corpus using the presented case restoration approach. For this purpose, we use an existing QA system as the baseline in experiments; we are not concerned with enhancing the QA system itself. A preprocessing module is designed to restore the case-sensitive form to feed to this QA system. The case restoration module is based on a Hidden Markov Model (HMM) trained on a large raw corpus of case sensitive documents, which are drawn from a given domain with no need for human annotation. With the plug-in of this preprocessing module, the entire QA system with its underlying NLP/IE components needs no change or adaptation in handling the case insensitive corpus. Using the TREC corpus with the case information artificially removed, this approach has been benchmarked with very good results, leading to only 2.8% degradation in QA performance. In the literature, this is the first time a QA system is applied to case insensitive corpora.

Although the artificially-made case insensitive corpus is an easier case than some real life corpora from speech recognition, the insight and techniques gained in this research are helpful in further exploring solutions of spoken language QA. In addition, by using the TREC corpus and the TREC benchmarking standards, the QA degradation benchmarking is easy to interpret and to compare with other QA systems in the community.

The case restoration approach has the following advantages: (i) the training corpus is almost limitless, resulting in a high performance model, with no knowledge bottleneck as faced by many supervised learning scenarios, (ii) the case restoration approach is applicable no matter whether the core system is statistical model, a hand-crafted rule system or a hybrid, (iii) when the core system consists of multiple modules, as is the case for the QA system used in the experiments that is based on multi-level NLP/IE, the case restoration approach relieves the burden of having to re-train or adapt each module in respect of case insensitive input, and (iv) the restoration approach reduces the system complexity: the burden of handling degraded text (*case* in this case) is reduced to a preprocessing module while all other components need no changes.

The remaining text is structured as follows. Section 2 presents the QA system. Section 3 describes the language model for case restoration. Section 4 benchmarks the IE engine and Section 5 benchmarks the IE-supported QA application. In both benchmarking sections, we compare the performance degradation from case sensitive input to case insensitive input. Section 5 is the Conclusion.

2 Question Answering Based on IE

We use a QA system supported by increasingly sophisticated levels of IE [Srihari & Li 2000] [Li *et al.* 2002]. Figure 1 presents the underlying IE engine *InfoXtract* [Srihari *et al.* 2003] that forms the basis for the QA system. The major information objects extracted by InfoXtract include NEs,[1] Correlated Entity (CE) relationships (e.g. *Affiliation*, *Position* etc.), Subject-Verb-Object (SVO) triples, entity profiles, and general or predefined events. These information objects capture the key content of the processed text, preparing a foundation for answering factoid questions.

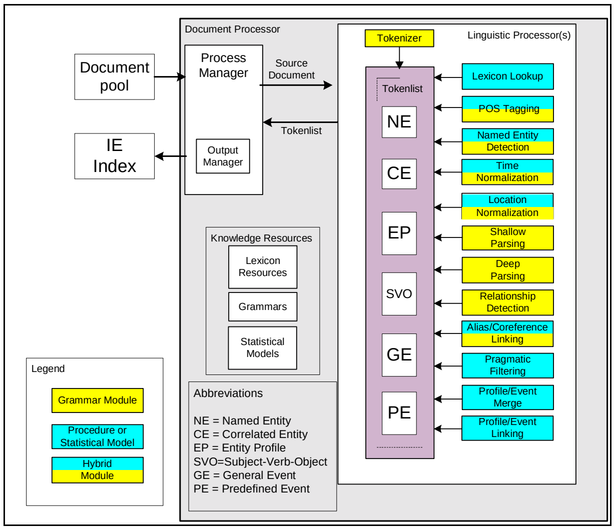

Figure 1: System Architecture of InfoXtract

Figure 2 shows the architecture of the QA system. This system consists of three components: (i) Question Processing, (ii) Text Processing, and (iii) Answer Ranking. In text processing, the case insensitive corpus is first pre-processed for case restoration before being parsed by InfoXtract. In addition, keyword indexing on the corpus is required. For question processing, a special module for Asking Point Identification is called for.

Linking the two processing components is the Answer Ranking component that consists of two modules: Snippet Retrieval and Feature Ranking.[2] Answer Ranking relies on access to information from both the Keyword Index as well as the IE Index.

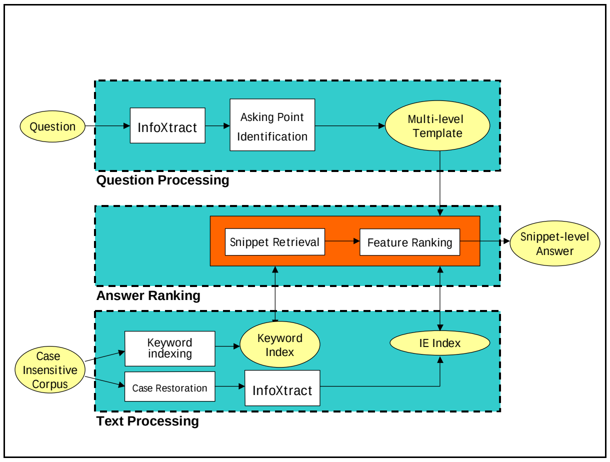

Figure 2: Architecture of QA Based on NLP/IE

Snippet Retrieval

Snippet retrieval generates the top *n* (we chose 200) most relevant sentence-level candidate answer snippets based on the question processing results.

We use two types of evidence for snippet retrieval: (i) keyword occurrence statistics at snippet level (with stop words removed), and (ii) the IE results, including NE Asking Points, Asking Point CE Link, head word of a phrase, etc.

If the Question Processing component detects an Asking Point CE Link, the system first attempts to retrieve snippets that contain the corresponding CE relationship. If it fails, it backs off to the corresponding NE Asking Point. This serves as a filter in the sense that only the snippets that contain at least one NE that matches the NE Asking Point are extracted. For questions that do not contain NE Asking Points, the system backs off to keyword-based snippet retrieval.

A synonym lexicon is also constructed for query expansion to help snippet retrieval. This includes irregular verbs (*go/went/gone*, etc.), verb-noun conversion (*develop/development*; *satisfy/satisfaction*; etc.), and a human-modified

[1] It is worth noting that there are two types of NE: (i) proper names *NeName* (including *NePerson*, *NeOrganization*, *NeLocation*, etc.) and (ii) non-name NEs (*NeItem*) such as time NE (*NeTimex*) and numerical NE (*NeNumex*). Close to 40% of the NE questions target non-name NEs. Proper name NEs are more subject to the case effect because recognizing a name in the running text often requires case information. Non-name NEs generally appear in predictable patterns. Pattern matching rules that perform case-insensitive matching are most effective in capturing them.

[2] There is a third, optional module Answer Point Identification in our QA system [10], which relies on deep parsing for generating phrase-level answers from snippet-level answers. This module was not used in the experiments reported in this paper.

conservative synonym list (e.g. *adjust/adapt; adjudicate/judge*; etc.).

Factors that contribute to relevancy weighting in snippet retrieval include giving more weight to the head words of phrases (e.g. 'disaster' in the noun phrase 'the costliest disaster'), more weight to words that are linked with question words (e.g. 'calories' in 'How many calories…' and 'American' in 'Who was the first American in space'), and discounting the weight for synonym matching.

Feature Ranking

The purpose of Feature Ranking is to re-rank the candidate snippets based on a list of ranking features.

Given a list of top *n* snippets retrieved in the previous stage, the Feature Ranking module uses a set of re-ranking features to fine-tune relevancy measures of the initial list of snippets in order to generate the final top five answer strings that are required as output. Figure 3 gives the ranking model for the Feature Ranking module.

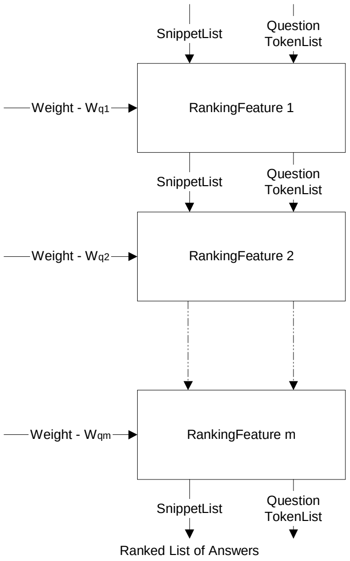

Figure 3: Pipeline for Ranking Features

For a given question, *Q*, let $\{S_1, S_2, \ldots, S_n\}$ be the set of candidate answer snippets. Let $\{R_1, R_2, \ldots, R_k\}$ be the ranking features. For a snippet S_j, let the ranking feature R_i assign a relevancy of r_{ij} quantifying the snippet's relevance to the question. The ranking model is given by

$$R(Q, S_j) = \sum_{i=1}^{k} w_{il} r_{ij}$$

where *l* represents the question type of *Q* and w_{il} gives the weight assigned to the ranking feature. Weights w_{il} vary based on question type.

We use both traditional IR ranking features such as Keyword Proximity and Inverse Document Frequency (IDF) as well as the ranking features supported by NLP/IE, listed below:

- NE Asking Point
- Asking Point CE Link
- Headword Match for Basic Phrases
- Phrase-Internal Word Order
- Alias (e.g. 'IBM' and 'International Business Machine')
- NE Hierarchical Match (e.g. *Company* vs. *Organization*)
- Structure-Based Matching (SVO Links, Head-Modifier Link, etc.)

3 Case Restoration

This section presents the case restoration approach [Niu *et al.* 2003] that supports QA on case insensitive corpus. The flowchart for using Case Restoration as a plug-in preprocessing module to IE is shown in Figure 4.

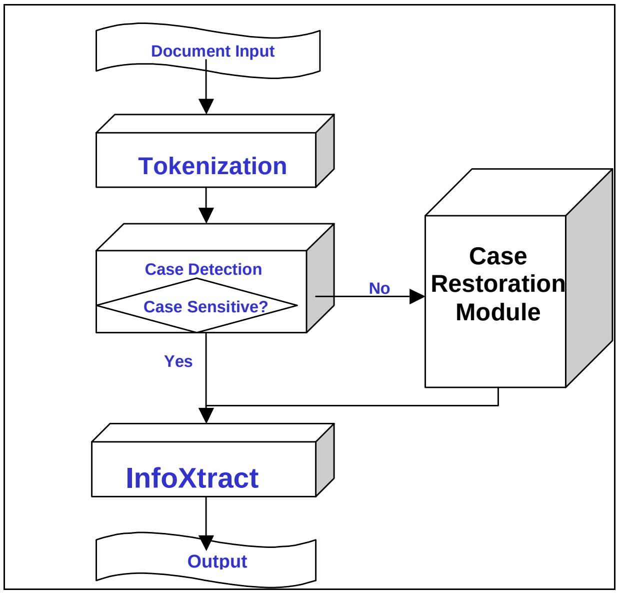

Figure 4: Case Restoration for IE

The incoming documents first go through tokenization. In this process, the case information is recorded as features for each token. This token-based case information provides basic evidence for the optional procedure called Case Detection to decide whether the Case Restoration module needs to be called.

A simple bi-gram Hidden Markov Model [Bikel *et al.* 1999] is selected as the choice of language model for this task. Currently, the system is based on a bi-gram model trained on a normal, case sensitive raw corpus in the chosen domain.

Three orthographic tags are defined in this model: (i) initial uppercase followed by at least one lowercase, (ii) all lowercase, and (iii) all uppercase.

To handle words with low frequency, each word is associated with one of five features: (i) *PunctuationMark* (e.g. &, ?, !…), (ii) *LetterDot* (e.g. A., J.P., U.S.A.,…), (iii) *Number* (e.g. 102,…), (iv) *Letters* (e.g. GOOD, MICROSOFT, IBM, …), or (v) *Other*.

The HMM is formulated as follows. Given a word sequence $W = \langle w_0 f_0 \rangle \cdots \langle w_n f_n \rangle$ (where f_j denotes a single token feature which are defined as above), the goal for the case restoration task is to find the optimal tag sequence $T = t_0 t_1 t_2 \cdots t_n$, which maximizes the conditional probability $Pr(T \mid W)$ [Bikel *et al.* 1999]. By Bayesian equality, this is equivalent to maximizing the joint probability $Pr(W,T)$. This joint probability can be computed by a bi-gram HMM as $Pr(W,T) = \prod_i Pr(\langle w_i, f_i \rangle, t_i \mid \langle w_{i-1}, f_{i-1} \rangle, t_{i-1})$. The back-off model is as follows,

$$\begin{aligned}
&\Pr(\langle w_i, f_i \rangle, t_i \mid \langle w_{i-1}, f_{i-1} \rangle, t_{i-1}) \\
&= \lambda_1 P_0(\langle w_i, f_i \rangle, t_i \mid \langle w_{i-1}, f_{i-1} \rangle, t_{i-1}) \\
&+ (1 - \lambda_1)\Pr(\langle w_i, f_i \rangle \mid t_i, t_{i-1})\Pr(t_i \mid w_{i-1}, t_{i-1}) \\
&\Pr(\langle w_i, f_i \rangle \mid t_i, t_{i-1}) \\
&= \lambda_2 P_0(\langle w_i, f_i \rangle \mid t_i, t_{i-1}) + (1 - \lambda_2)\Pr(\langle w_i, f_i \rangle \mid t_i) \\
&\Pr(t_i \mid w_{i-1}, t_{i-1}) \\
&= \lambda_3 P_0(t_i \mid w_{i-1}, t_{i-1}) + (1 - \lambda_3)\Pr(t_i \mid w_{i-1}) \\
&\Pr(\langle w_i, f_i \rangle \mid t_i) \\
&= \lambda_4 P_0(\langle w_i, f_i \rangle \mid t_i) + (1 - \lambda_4)\Pr(w_i \mid t_i)P_0(f_i \mid t_i) \\
&\Pr(t_i \mid w_{i-1}) = \lambda_5 P_0(t_i \mid w_{i-1}) + (1 - \lambda_5)P_0(t_i) \\
&\Pr(w_i \mid t_i) = \lambda_6 P_0(w_i \mid t_i) + (1 - \lambda_6)\frac{1}{V}
\end{aligned}$$

where V denotes the size of the vocabulary, the back-off coefficients λ's are determined using the Witten-Bell smoothing algorithm, and the quantities $P_0(\langle w_i, f_i \rangle, t_i \mid \langle w_{i-1}, f_{i-1} \rangle, t_{i-1})$, $P_0(\langle w_i, f_i \rangle \mid t_i, t_{i-1})$, $P_0(t_i \mid w_{i-1}, t_{i-1})$, $P_0(\langle w_i, f_i \rangle \mid t_i)$, $P_0(f_i \mid t_i)$, $P_0(t_i \mid w_{i-1})$, $P_0(t_i)$, and $P_0(w_i \mid t_i)$ are computed by the maximum likelihood estimation.

A separate HMM is trained for bigrams involving unknown words. The training corpus is separated into two parts, the words occurring in Part I but not in Part II and the words occurring in Part II but not in Part I are all replaced by a special symbol *#Unknown#*. Then an HMM for unknown words is trained on this newly marked corpus. In the stage of tagging, the unknown word model is used in case a word beyond the vocabulary occurs.

4 IE Engine Benchmarking

A series of benchmarks have been conducted in evaluating the approach presented in this paper. They indicate that this is a simple but very effective method to solve the problem of handling case insensitive input for NLP, IE and QA.

Case Restoration

A raw corpus of 7.6 million words in mixed case drawn from the general news domain is used in training case restoration. A separate testing corpus of 0.88 million words drawn from the same domain is used for benchmarking. Table 1 gives the case restoration performance benchmarks. The overall F-measure is 98% (P for Precision, R for Recall and F for F-measure).

Table 1: Case Restoration Performance

			P	R	F
Overall			0.96	1	0.98
	Lower Case		0.97	0.99	0.98
	Non-Lower Case		0.93	0.84	0.88
		Initial-Upper Case	0.87	0.84	0.85
		All-Upper Case	0.77	0.6	0.67

The score that is most important for IE is the F-measure of recognizing non-lowercase word. We found that the majority of errors involve missing the first word in a sentence due to the lack of a powerful sentence final punctuation detection module in the case restoration stage. But it is found that such ‘errors’ have almost no negative effect on the following IE tasks.[3]

There is no doubt that the lack of case information from the input text will impact the NLP/IE/QA performance. The goal of the case restoration module is to minimize this impact. A series of degradation tests have been run to measure the impact.

Degradation Tests on IE and Parsing

Since IE is the foundation for our QA system, the IE degradation due to the case insensitive input directly affects the QA performance.

The IE degradation benchmarking is designed as follows. We start with a testing corpus drawn from normal case sensitive text. We then feed the corpus into the IE engine for benchmarking. This is normal benchmarking for case sensitive text input as a baseline. After that, we artificially remove the case information by transforming the corpus into a corpus in all uppercase. The case restoration module is then plugged in to restore the case before feeding the IE engine. By comparing benchmarking using case restoration with baseline benchmarking, we can calculate the level of performance degradation from the baseline in handling case insensitive input.

For NE, an annotated testing corpus of 177,000 words is used for benchmarking (Table 3), using an automatic scorer following Message Understanding Conference (MUC) NE standards.

Table 2: NE Degradation Benchmarking

Type	P	R	F
NE on case sensitive input	89.1%	89.7%	89.4%
NE on case insensitive input using case restoration	86.8%	87.9%	87.3%
Degradation	2.3%	1.8%	**2.1%**

The overall F-measure for NE degradation, due to the loss of case information in the incoming corpus, is 2.1%. We have also implemented the traditional NE-retraining approach proposed by [Kubala *et al*. 1998] [Miller *et al.* 2000] [Palmer *et al.* 2000] and the re-trained NE model leads to 6.3% degradation in the NE F-measure, a drop of more than four percentage points when compared with the case restoration two-step approach. Since this comparison between two approaches is based on the same testing corpus using the same system, the conclusion can be derived that the case restoration approach is clearly better than the retraining approach for NE.

Beyond NE, some fundamental InfoXtract support for QA comes from the CE relationships and the SVO parsing results. We benchmarked their degradation as follows.

From a processed corpus drawn from the news domain, we randomly picked 250 SVO structural links and 60 AFFILIATION and POSITION relationships for manual checking (Table 3, COR for Correct, INC for Incorrect, SPU for Spurious, MIS for Missing, and DEG for Degradation).

Surprisingly, there is almost no statistically significant difference in the SVO performance. The degradation due to the case restoration was only 0.07%. This indicates that parsing is less subject to the case factor to a degree that the performance differences between a normal case sensitive input and a case restored input are not obviously detectable.

Table 3: SVO/CE Degradation Benchmarking

	SVO			CE		
	Baseline	**Case Restored**		**Baseline**	**Case Restored**	
COR	196	195	**DEG**	48	43	**DEG**
INC	13	12		0	1	
SPU	10	10		2	2	
MIS	31	33		10	14	
P	89.50%	89.86%	-0.36%	96.0%	93.5%	2.5%
R	81.67%	81.25%	0.42%	82.8%	74.1%	8.7%
F	**85.41%**	**85.34%**	**0.07%**	**88.9%**	**82.7%**	**6.2%**

The degradation for CE is about 6%. Considering there is absolutely no adaptation of the CE module, this degradation is reasonable.

5 QA Degradation Benchmarking

The QA experiments were conducted following the TREC-8 QA standards in the category of 250-byte answer strings. In addition to the TREC-8 benchmarking standards Mean Reciprocal Rank (MRR), we also benchmarked precision for the top answer string (Table 4).

[3] In fact, positive effects are observed in some cases. The normal English orthographic rule that the first word be capitalized can confuse the NE learning system due to the lack of the usual orthographic distinction between a candidate proper name and a common word.

Table 4: QA Degradation Benchmarking-1

Type	Top 1 Precision	MRR
QA on case sensitive corpus	130/198=65.7%	73.9%
QA on case insensitive corpus	124/198=62.6%	71.1%
Degradation	**3.1%**	**2.8%**

Comparing QA benchmarks with benchmarks for the underlying IE engine shows that the limited QA degradation is in proportion with the limited degradation in NE, CE and SVO. The following examples illustrate the chain effect: case restoration errors → NE/CE/SVO errors → QA errors.

Q137: 'Who is the mayor of Marbella?'

This is a CE question, the decoded CE asking relationship is *CeHead* for the location entity 'Marbella'. In QA on the original case sensitive corpus, the top answer string has a corresponding CeHead relationship extracted as shown below.

Input: Some may want to view the results of the much-publicised activities of the mayor of Marbella, Jesus Gil y Gil, in cleaning up the town

→ [NE tagging]

Some may want to view the results of the much-publicised activities of the mayor of <NeCity>Marbella</NeCity> , <NeMan>Jesus Gil y Gil</NeMan>, in cleaning up the town

→ [CE extraction]

CeHead: Marbella → Jesus Gil y Gil

In contrast, the case insensitive processing is shown below:

Input: SOME MAY WANT TO VIEW THE RESULTS OF THE MUCH-PUBLICISED ACTIVITIES OF THE MAYOR OF MARBELLA, JESUS GIL Y GIL, IN CLEANING UP THE TOWN

→ [case restoration]

some may want to view the results of the much-publicised activities of the mayor of marbella , Jesus Gil y Gil, in cleaning up the town

→ [NE tagging]

some may want to view the results of the much-publicised activities of the mayor of marbella , <NeMan>Jesus Gil y Gil</NeMan> , in cleaning up the town

The CE module failed to extract the relationship for MARBELLA because this relationship is defined for the entity type NeOrganization or NeLocation which is absent due to the failed case restoration for 'MARBELLA'. The next example shows an NE error leading to a problem in QA.

Q119: 'What Nobel laureate was expelled from the Philippines before the conference on East Timor?'

In question processing, the NE Asking Point is identified as NePerson. Because *Mairead Maguire* was successfully tagged as NeWoman, the QA system got the correct answer string in the following snippet: *Immigration officials at the Manila airport on Saturday expelled Irish Nobel peace prize winner Mairead Maguire*. However, the case insensitive processing fails to tag any NePerson in this snippet. As a result the system misses this answer string. The process is illustrated below.

Input: IMMIGRATION OFFICIALS AT THE MANILA AIRPORT ON SATURDAY EXPELLED IRISH NOBEL PEACE PRIZE WINNER MAIREAD MAGUIRE

→ [case restoration]

immigration officials at the Manila airport on Saturday expelled Irish Nobel Peace Prize Winner Mairead Maguire

→ [NE tagging]

immigration officials at the <NeCity>Manila</NeCity> airport on <NeDay>Saturday</NeDay> expelled <NeProduct>Irish Nobel Peace Prize Winner Mairead Maguire </NeProduct>

As shown, errors in case restoration cause mistakes in the NE grouping and tagging: *Irish Nobel Peace Prize Winner Mairead Maguire* is wrongly tagged as NeProduct.

We also found one interesting case where case restoration actually leads to QA performance enhancement over the original case sensitive processing. A correct answer snippet is promoted from the 3rd candidate to the top in answering Q191 'Where was Harry Truman born?'. This process is shown below.

Input: HARRY TRUMAN (33RD PRESIDENT): BORN MAY 8, 1884, IN LAMAR, MO.

→ [case restoration]
Harry Truman (33rd President) : born May 8 , 1884 , in Lamar , MO .
→ [NE tagging]
<NeMan>Harry Truman</NeMan> (<NeOrdinal>33rd</NeOrdinal> President) : born <NeDay>May 8 , 1884</NeDay> , in <NeCity>Lamar , MO</NeCity> .

As shown, *LAMAR, MO* gets correctly tagged as NeCity after case restoration. But *LAMAR* is mis-tagged as NeOrg in the original case sensitive processing. The original case sensitive snippet is *Harry Truman (33rd President): Born May 8, 1884, in Lamar, Mo.* In our NE system, there is such a learned pattern as follows:

X , TwoLetterUpperCase → NeCity.

This rule fails to apply to the original text because the US state abbreviation appears in a less frequently seen format *Mo* instead of *MO*. However, the restoration HMM assigns all uppercase to 'MO' since this is the most frequently seen orthography for this token. This difference of the restored case from the original case enables the NE tagger to tag *Lamar, MO* as 'NeCity' which meets the NE Asking Point constraint 'NeLocation'.

QA and Case Insensitive Question

We also conducted a test on case insensitive questions in addition to case insensitive corpus by calling the same case restoration module.

Table 5: QA Degradation Benchmarking-2

Type	Top 1 Precision	MRR
QA on case sensitive corpus	130/198=65.7%	73.9%
QA on case insensitive corpus, with case insensitive question	111/198=56.1%	64.4%
Degradation	**9.6%**	**9.5%**

This research is useful because, when interfacing a speech recognizer to a QA system to accept spoken questions, the case information is not available in the incoming question.[4] We want to know how the same case restoration technique applies to question processing and gauge the degradation effect on the QA performance (Table 5).

We notice that the question processor missed two originally detected NE Asking Points and one Asking Point CE Link. There are a number of other errors due to incorrectly restored case, including non-asking-point NEs in the question and grouping errors in shallow parsing as shown below for Q26 : 'What is the name of the "female" counterpart to El Nino, which results in cooling temperatures and very dry weather?' (Notation: NP for Noun Phrase, VG for Verb Group, PP for Prepositional Phrase and AP for Adjective Phrase).

Input: WHAT IS THE NAME OF THE "FEMALE" COUNTERPART TO EL NINO … ?
→ [case restoration]
What is the name of the "Female" counterpart to El Nino, …?
→ [question shallow parsing]
NP[What] VG[is] NP[the name] PP[of the] " AP[Female] " NP[counterpart] PP[to El Nino] , … ?

In the original mixed-case question, after parsing, we get the following basic phrase grouping:

NP[What] VG[is] NP[the name] PP[of the " female " counterpart] PP[to El Nino] , … ?

There is only one difference between the case-restored question and the original mixed-case question, i.e. *Female* vs. *female*. This difference causes the shallow parsing grouping error for the PP *of the "female" counterpart*. This error affects the weights of the ranking features Headword Matching and Phrase-internal Word Order. As a result, the following originally correctly identified answer snippet was dropped: *the greenhouse effect and El Nino -- as well as its "female" counterpart, La Nina -- have had a profound effect on weather nationwide.*

As question processing results are the starting point and basis for snippet retrieval and feature ranking, an error in question processing seems to lead to greater degradation, as seen in almost 10% drop compared with about 3% drop in the case when only the corpus is case insensitive.

[4] In addition to missing the case information, there are other aspects of spoken questions that require treatment, e.g., lack of punctuation marks, spelling mistakes, repetitions. Whether the restoration approach is effective calls for more research.

A related explanation for this degradation contrast is as follows. Due to the information redundancy in a large corpus, processing errors in some potential answer strings in the corpus can be compensated for by correctly processed equivalent answer strings. This is due to the fact that the same answer may be expressed in numerous ways in the corpus. Some of those ways may be less subject to the case effect than others. Question processing errors are fatal in the sense that there is no information redundancy for its compensation. Once it is wrong, it directs the search for answer strings in the wrong direction. Since questions constitute a subset of the natural language phenomena with their own characteristics, case restoration needs to adapt to this subset for optimal performance, e.g. by including more questions in the case restoration training corpus.

6 Conclusion

An effective approach to perform QA on case insensitive corpus is presented with very little degradation (2.8%). This approach uses a high performance case restoration module based on HMM as a preprocessor for the NLP/IE processing of the corpus. There is no need for any changes on the QA system and the underlying IE engine which were originally designed for handling normal, case sensitive corpora. It is observed that the limited QA degradation is due to the limited IE degradation.

An observation from the research of handling case insensitive questions is that question processing degradation has more serious consequence affecting the QA performance. The current case restoration training corpus is drawn from the general news articles which rarely contain questions. As a future effort, we plan to focus on enhancing the case restoration performance by including as many mixed-case questions as possible into the training corpus for case restoration.

Acknowledgment

This work was partly supported by a grant from the Air Force Research Laboratory's Information Directorate (AFRL/IF), Rome, NY, under contract F30602-03-C-0044. The authors wish to thank Carrie Pine and Sharon Walter of AFRL for supporting and reviewing this work.

References

Abney, S., Collins, M and Singhal. 2000. A. Answer Extraction. *Proceedings of ANLP-2000*, Seattle.

Bikel, D.M. et al. 1997. Nymble: a High-Performance Learning Name-finder. *Proceedings of the Fifth Conference on ANLP*, Morgan Kaufmann Publishers, 194-201.

Bikel, D.M., R. Schwartz, and R.M. Weischedel. 1999. An Algorithm that Learns What's in a Name. *Machine Learning*, Vol. 1,3, 1999, 211-231.

Chieu, H.L. and H.T. Ng. 2002. Teaching a Weaker Classifier: Named Entity Recognition on Upper Case Text. *Proceedings of ACL-2002*, Philadelphia.

Chinchor N. and E. Marsh. 1998. MUC-7 Information Extraction Task Definition (version 5.1), *Proceedings of MUC-7*.

Hovy, E.H., U. Hermjakob, and Chin-Yew Lin. 2001. The Use of External Knowledge of Factoid QA. Proceedings of TREC-10, 2001, Gaithersburg, MD, U.S.A..

Krupka, G.R. and K. Hausman. 1998. IsoQuest Inc.: Description of the NetOwl (TM) Extractor System as Used for MUC-7, *Proceedings of MUC-7*.

Kubala, F., R. Schwartz, R. Stone and R. Weischedel. 1998. Named Entity Extraction from Speech. *Proceedings of DARPA Broadcast News Transcription and Understanding Workshop*.

Kupiec J. 1993. MURAX: A Robust Linguistic Approach For Question Answering Using An On-Line Encyclopaedia. *Proceedings of SIGIR* Pittsburgh, PA.

Li, W, R. Srihari, X. Li, M. Srikanth, X. Zhang and C. Niu. 2002. Extracting Exact Answers to Questions Based on Structural Links. *Proceedings of Multilingual Summarization and Question Answering* (COLING-2002 Workshop), Taipei, Taiwan.

Litkowski, K. C. 1999. Question-Answering Using Semantic Relation Triples. *Proceedings of TREC-8*, Gaithersburg, MD.

Miller, D., S. Boisen, R. Schwartz, R. Stone, and R. Weischedel. 2000. Named Entity Extraction from Noisy Input: Speech and OCR. *Proceedings of ANLP 2000*, Seattle.

Niu, C., W. Li, J. Ding and R. Srihari. 2003. Orthographic Case Restoration Using Supervised Learning Without Manual Annotation. *Proceedings of the 16th International FLAIRS Conference 2003*, Florida

Chincor, N., P. Robinson and E. Brown. 1998. *HUB-4 Named Entity Task Definition Version 4.8*. (www.nist.gov/speech/tests/bnr/hub4_98/ hub4_98.htm)

Palmer, D., M. Ostendorf and J.D. Burger. 2000. Robust Information Extraction from Automatically Generated Speech Transcriptions. *Speech Communications*, Vol. 32, 2000, 95-109.

Pasca, M. and S.M. Harabagiu. 2001. High Performance Question/Answering. *Proceedings of SIGIR 2001*, 366-374.

Robinson, P., E. Brown, J. Burger, N. Chinchor, A. Douthat, L. Ferro, and L. Hirschman. 1999. Overview: Information Extraction from Broadcast News. *Proceedings of The DARPA Broadcast News Workshop* Herndon, Virginia.

Srihari, R and W. Li. 2000. A Question Answering System Supported by Information Extraction. *Proceedings of ANLP 2000*, Seattle.

Srihari, R., W. Li, C. Niu and T. Cornell. 2003. InfoXtract: A Customizable Intermediate Level Information Extraction Engine. *HLT-NAACL03 Workshop on The Software Engineering and Architecture of Language Technology Systems (SEALTS).* Edmonton, Canada

Voorhees, E. 1999. The TREC-8 Question Answering Track Report. *Proceedings of TREC-8*. Gaithersburg, MD.

Voorhees, E. 2000. Overview of the TREC-9 Question Answering Track. *Proceedings of TREC-9*. Gaithersburg, MD.